;File: tic41val.a51

;Author: Eckart Hartmann Date:15/10/2003

;Development progress: Tic834.df

;

;TicVal==========Set and start interval counter.

;C Function prototype: char TicVal(char cVal, char cMode);

;Description of Function: Put interval counter in operation.

;User interface: Set first parameter to required interval count.

; Set second parameter to choose clock source for interval counter

; as follows:

; 0 for 128 clocks per second (HTHSEC)

; 1 for 1 clocks per second (SEC)

; 2 for 1 clocks per minute (MIN)

; 3 for 1 clocks per hour (HOUR)

; Add 8 to the parameter for a single interval or else multiple

; intervals are measured.

; Call TicVal function.

; TicVal zeroes interval counter sets required interval count

; and clock source and starts interval counting.

; Enables global interupt bit and TIC interupt on low priority.

; When required interval is reached interupt TicInt is

; executed.

; returns TIMECON (guaranteed not zero).

;Robustness:

;Side effects: Starts TIC timer if not running already.

;

NAME TICVAL

$NOMOD51

$IC(..kei841.inc) ; Parameter passing registers for Keil .

$IC(..kei841.dat) ; SFR definition for Keil .

;

public \_TicVal

?PR?\_TicVal?TICVAL SEGMENT CODE

RSEG ?PR?\_TicVal?TICVAL

;

\_TicVal:

anl TIMECON,#0f1h ;Stop and clear interval counter.

mov a,cP2lc ;Choose single or multiple.

anl a,#8

orl TIMECON,a

mov a,cP2lc ;Choose clock source.

anl a,#03h

swap a

anl TIMECON,#0cfh

orl TIMECON,a

mov INTVAL,cP1l ;Set interval.

orl TIMECON,#3 ;Start TIC.

anl IEIP2,#0afh ;TIC low priority

orl IEIP2,#4 ; enabled.

setb EA ;Global enable.

mov cP1l,TIMECON ;Return TIMECON.

ret

;

;Function End==========================================================Function End

END